IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of	)	
	)	
Eckhard Kruse et al.	)	Group Art Unit: 2442
	)	
Application No.: 10/562,046	)	Examiner: Jeffrey L. Nickerson
	)	
Filed: April 11, 2006	)	Appeal No.: _____
	)	
For: METHOD AND SYSTEM FOR	)	
EVENT TRANSMISSION	)	
	)	
	)	
	)	

RESPONSE TO NOTIFICATION OF NON-COMPLIANT APPEAL BRIEF

Mail Stop APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Notification dated November 10, 2009, Appellants submit herewith a replacement section V (Summary of Claimed Subject Matter) and replacement section VIII (Claims Appendix) to address the objections to the Brief under 37 CFR 41.37(c)(1) (v), (viii).

Pursuant to Paragraph "4(a)" and "10" of the Notification, the Brief has been revised to more explicitly contain a concise explanation of the subject matter defined in each independent claim, with reference to the specification by page and line number and to the drawings by reference characters.

Pursuant to Paragraph "4(b)" for each of independent claims 21 and 27, the Brief does identify structure, material or acts described in the specification as corresponding to teach claimed function, with reference to the specification by page and line number and to the drawings. Contrary to Paragraph 10 of the Notification, the independent claims are 21 and 27, not 26, 27. However, because claim 26 is also on appeal, functions of this claim have been mapped to the specification and drawings in the claim chart included with the Summary Of Claimed Subject Matter.

Pursuant to Paragraph 7 of the Notification, a revised copy of the appealed claims is submitted wherein the objection to claim 36 in paragraph "10" of the Notification has been addressed.

Buchanan Ingersoll ⚭ Rooney PC
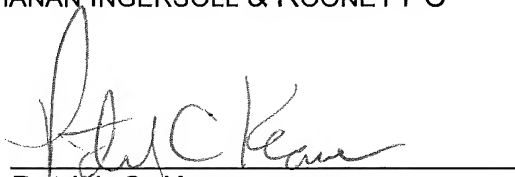Attorneys & Government Relations Professionals

The Commissioner is hereby authorized to charge any appropriate fees under 37 C.F.R. §§1.16, 1.17, and 1.21 that may be required by this paper, and to credit any overpayment, to Deposit Account No. 02-4800.

Respectfully submitted,

BUCHANAN INGERSOLL & ROONEY PC

Date December 10, 2009   By: _____

Patrick C, Keane
Registration No. 32,858

P.O. Box 1404
Alexandria, VA 22313-1404
703 836 6620

V.   Summary Of Claimed Subject Matter

Exemplary embodiments are directed to methods for monitoring and/or controlling technical installations, such as production installations. A client application can be used as an operator interface to perform installation monitoring and control. Events which occur in the technical installation, such as alarm conditions, are supplied via a data capture unit connected to a server.

Appellants' claims include independent claims <u>21 and 27</u>.

<u>Claim 21:</u>   Appellant' <u>claim 21</u> encompasses a method whereby a client application need not request event transmission from the server, and a correspondence of claim features to exemplary embodiments described in the specification and drawings is set forth in the claim chart below. Appellants have disclosed a client application which communicates with a "client event service" whereby events can be "logged" by the client event service and an associated "server event service", such that the events can be transmitted to the client application over a communication link between the server and a client <u>without any active request from the client application</u>. As such the client application can function independently of the server.

Referring to specification page 11, lines 21-24 and to Appellants' exemplary Figure 2, a system is illustrated which can manage and transmit events from a server 2 to a client application 4 by which the client can see event handling and data transmission initiated by the server as though they were occurring in a local environment of the client. For events to be transmitted from the server via the communication link 9 to the client application, the event is logged using the server event service 7 and the client event service 6. Events for which logging has been performed are transmitted from the server, via the client event service 6, to a client application 4 using callback functions and an event identifier. An event identifier is illustrated in Figure 3.

A first logging can prompt an initialization of the client/server system. When an event occurs in the Figure 2 system, it is reported to an installation interface 10. If the event in question has been logged, it is transferred from the installation interface to the server event service 7.

Thus, if there is an event that has been detected by the server event service, it is transmitted via the communication link to the client event service based on the request received from the <u>client event service</u> 6; <u>not</u> from the <u>client application</u> 4 to which it is ultimately transmitted. The client event service transmits received events to the client application 4, where the event is reported. By avoiding a need for active requests from the client application, the client application need not communicate with the server, and the client application can be independent of the server.

In Appellants' exemplary Figure 2, the client application 4 logs a client callback function 41 in the client event service 6 (see client logging function 61) for an event about which the client application 4 is to be notified. The client event service uses the communication link 9 to log a corresponding server callback function 72 in the server event service 7.

Callback functions can be logged for an event with which a same event name is associated with the client and with the server. The client application 4 can call a client logging function from the client event service and provide it with a name of an event in question and with a pointer to the client callback function which is to be logged. The client logging function can log a unique event identifier, and transmit this event identifier together with the event name to a server logging function 71 of the server event service 7. As such, the event name and unique event identifier can be associated with an event which occurs at the installation interface 10.

The server logging function 71 can log a server callback function 72 with the installation interface 10 by transferring the event name. The server logging function can use a server event table 74 to store a data record which contains the event identifier, and a pointer to the server callback function which is to be logged. The server logging function can use the communication link 9 to report back to the client logging function of the client event service that the logging operation has been performed.

<u>Claim 26:</u>   The dependent method claim 26 recites features relating to the transmission of events, and a reference of claim 26 features to the specification and drawings, is set forth in the claim chart below. An exemplary operation is described

at Specification page 15, line 37 to page 17, line 16 with reference to Figure 2 server 2, and client 1.

Claim 27: An exemplary system as encompassed by Appellants' claim 27 is described in the Specification and drawings in the claim chart below, and includes a client 1, and a client event service 6 to make requests for event transmission to a server event service 7 of a server 2 which has a server logging function 71. A server event table 74 is formed as a hash table and holds data records which contain at least one event identifier and a pointer to a server callback function which is to be logged. An event queue 75 can hold entries which describe a respective event, and transmit received events to a client application. Thus, the claim 27 system includes a "client event service" and a "server event service", such that received events can be transmitted to a separate client application.

21. A method for managing and transmitting events from a server via a communication link to at least one client, said method comprising:	E.g., Fig. 2, Server 2; client 1, communication link 9; Spec. p. 11, lines 1-6.
logging possible events in a client event service for the purpose of initializing or updating the client,	Client event service 6
logging possible events in a server event service for the purpose of initializing or updating the server,	Server event service 7
transferring detected events which have been logged from an installation interface to the server event service,	Installation interface 10
sending requests initiated by the client event service regarding the detected events to the server event service,	
transmitting the detected events to the client event service on the basis of a request which has been made to the server event service, and	
transmitting events received by	

the client event service to a client application,	
wherein the client application logs a client callback function in the client event service for every event about which it is to be notified, and the client event service uses the communication link to log a corresponding server callback function in the server event service, and	Client application 4 Client callback function 41 Server callback function 72
wherein to log the callback functions for an event with which a same event name is associated with the client and with the server, the following steps are performed:	E.g., Specification Page 14, lines 2-34
calling, by the client application, a client logging function from the client event service and providing said function with a name of an event in question and with a pointer to the client callback function which is to be logged,	Client application 4 Client logging function 61
logging, by the client logging function, a unique event identifier,	Unique event identifier (Fig. 3); E.g., Spec. p. 14, lls. 8-9
transmitting the event identifier and the event name via the communication link to a server logging function of the server event service,	Server logging function 71
logging, by the server logging function, a server callback function with the installation interface by transferring the event name,	Server callback function 72
storing, by the server logging function in a server event table, a data record, which contains at least the event identifier and a pointer to the server callback function which is to be logged,	Server event table 74 (see Fig. 3)
reporting, by the server logging function, performance of the logging operation to the client logging function of the client event service via the	

communication link, and logging, by the client logging function, the client callback function by storing a data record in a client event table, the data record containing at least the event identifier and a pointer to the client callback function which is to be logged.	Data record 73 Client event table 62 (see Fig. 3)
22. The method as claimed in claim 21, wherein the events to be transmitted are detected by a data capture unit in a technical installation and are reported to the installation interface of the server.	Installation interface 10
24. The method as claimed in claim 21, wherein after a client callback function has been logged for the first time the client logging function starts a request generator which then makes requests for event transmission to the server event service.	Request generator 63
25. The method as claimed in claim 24, wherein the request generator of the client event service makes the requests for event transmission to the server event service cyclically.	Specification, page 10, lines 3-11
26. The method as claimed in claim 24, wherein events are transmitted by performing the following steps: detecting, by the installation interface, an event which has occurred and calling the server callback function logged for this event, producing, by the server callback function, an entry describing the event in at least one event queue, reading, by the server event service, the entry produced in the event queue upon the next request from the client event service for event transmission,	Fig. 2, Server 2, client 1 Specification p. 15, line 37 to p. 17, line 16 Installation interface 10 Server callback function 72; Event queue 75 Server event server 7, client event server 6

transmitting the entry via the communication link to the client event service,	Communication link 10; Client event service 6
receiving, by the client event service, the entry,	Client event service 6
ascertaining and calling the client callback function logged for the event, and	
executing, by the client callback function, a defined action for the corresponding event in the client application.	Client callback function 41
27. A system for managing and transmitting events from a server via a communication link to at least one client, said system comprising:	E.g., Figs. 2, 3; server 2; client 1; communication link 9
at least one client, comprising:	client 1
at least one client event service, for logging possible events, which uses a communication link to make requests for event transmission to a server event service,	Client event service 6 Server event service 7
a server, comprising:	Server 2
at least one server event service, which has at least one server logging function for logging server callback functions, and for logging possible events, and which uses a communication link to transmit events to the client event service,	Server logging function 71 Server callback function 72
at least one server event table for holding data records which describe a respective logging operation, which server event table is formed as a hash table and holds data records which contain at least one event identifier and a pointer to a server callback function which is to be logged,	Server event table 74 Event identifier (Fig. 3) Pointer

at least one event queue for holding entries which describe a respective event, and for transmitting received events to a client application, and at least one installation interface which transfers events which have occurred to the at least one server event service.	Event queue 75 Client application 4 Installation interface 10
28. The method as claimed in claim 21, wherein optionally a tidying function of the server event service is called which deletes the server event table and an event queue if the client event service is no longer communicating with the server event service.	Tidying function 77
29. The system as claimed in claim 27, wherein the installation interface is connected to a data capture unit of a technical installation in order to read in events detected by the data capture unit.	Installation interface 10
30. The system as claimed in claim 27, wherein the server event service has at least one server callback function which can be logged for at least one event and which is called when an event for which it is logged occurs.	Server callback function 72
31. The system as claimed in claim 27, wherein the server event service has, for every client event service with which it communicates via a communication link, a separate client data record which respectively contains at least one server event table and at least one event queue.	Client data record 73 Server event table 74 Event queue 75
32. The system as claimed in claim 31, wherein the server event service has a tidying function which deletes the client data record if the associated client event service is no longer communicating with	Tidying function 77

the server event service.	
33. The system as claimed in claim 31, wherein the server event table is in the form of a hash table and holds data records which contain at least one event identifier and a pointer to a server callback function which is to be logged.	Server event table 74
34. The system as claimed in claim 27, wherein the client event service has at least one client logging function for logging client callback functions, at least one client event table for holding data records which describe the log, and at least one request generator for making cyclic requests for event transmission.	Client logging function 61

Client event table 62

Request generator 63 |
35. The system as claimed in claim 34, wherein the client event table is in the form of a hash table and holds data records which contain at least one event identifier and a pointer to a client callback function which is to be logged.	Client event table 62
36. The method as claimed in claim 21, wherein events are transmitted by performing the following steps:

        detecting, by the installation interface, an event which has occurred and calling the server callback function logged for this event,

        producing, by the server callback function, an entry describing the event in at least one event queue,

        reading, by the server event service, the entry produced in the event queue upon the next request from the client event service for event transmission, | Installation interface 10;
Server callback function 72

Event queue 75

Server event service 7

Client event service 6 |

transmitting the entry via the communication link to the client event service, receiving, by the client event service, the entry, ascertaining and calling the client callback function logged for this event, and executing, by the client callback function, a defined action for the corresponding event in the client application.	Client callback function 41

VIII.   Claims Appendix

See attached Claims Appendix for a copy of the claims involved in the appeal.

VIII. CLAIMS APPENDIX

The Appealed Claims

1-20. (Canceled)

21. A method for managing and transmitting events from a server via a communication link to at least one client, said method comprising:

logging possible events in a client event service for the purpose of initializing or updating the client,

logging possible events in a server event service for the purpose of initializing or updating the server,

transferring detected events which have been logged from an installation interface to the server event service,

sending requests initiated by the client event service regarding the detected events to the server event service,

transmitting the detected events to the client event service on the basis of a request which has been made to the server event service, and

transmitting events received by the client event service to a client application,

wherein the client application logs a client callback function in the client event service for every event about which it is to be notified, and the client event service uses the communication link to log a corresponding server callback function in the server event service, and

wherein to log the callback functions for an event with which a same event name is associated with the client and with the server, the following steps are performed:

calling, by the client application, a client logging function from the client event service and providing said function with a name of an event in question and with a

pointer to the client callback function which is to be logged,

logging, by the client logging function, a unique event identifier,

transmitting the event identifier and the event name via the communication

link to a server logging function of the server event service,

logging, by the server logging function, a server callback function with the

installation interface by transferring the event name,

storing, by the server logging function in a server event table, a data record,

which contains at least the event identifier and a pointer to the server callback

function which is to be logged,

reporting, by the server logging function, performance of the logging operation

to the client logging function of the client event service via the communication link,

and

logging, by the client logging function, the client callback function by storing a

data record in a client event table, the data record containing at least the event

identifier and a pointer to the client callback function which is to be logged.

22. The method as claimed in claim 21, wherein the events to be transmitted are

detected by a data capture unit in a technical installation and are reported to the

installation interface of the server.

23. (Canceled)

24. The method as claimed in claim 21, wherein after a client callback function has

been logged for the first time the client logging function starts a request generator

which then makes requests for event transmission to the server event service.

25. The method as claimed in claim 24, wherein the request generator of the client event service makes the requests for event transmission to the server event service cyclically.

26. The method as claimed in claim 24, wherein events are transmitted by performing the following steps:

detecting, by the installation interface, an event which has occurred and calling the server callback function logged for this event,

producing, by the server callback function, an entry describing the event in at least one event queue,

reading, by the server event service, the entry produced in the event queue upon the next request from the client event service for event transmission,

transmitting the entry via the communication link to the client event service,

receiving, by the client event service, the entry,

ascertaining and calling the client callback function logged for the event, and

executing, by the client callback function, a defined action for the corresponding event in the client application.

27. A system for managing and transmitting events from a server via a communication link to at least one client, said system comprising:

at least one client, comprising:

at least one client event service, for logging possible events, which uses a communication link to make requests for event transmission to a server event service,

a server, comprising:

at least one server event service, which has at least one server logging function for logging server callback functions, and for logging possible events, and which uses a communication link to transmit events to the client event service,

at least one server event table for holding data records which describe a respective logging operation, which server event table is formed as a hash table and holds data records which contain at least one event identifier and a pointer to a server callback function which is to be logged,

at least one event queue for holding entries which describe a respective event, and for transmitting received events to a client application, and

at least one installation interface which transfers events which have occurred to the at least one server event service.

28. The method as claimed in claim 21, wherein optionally a tidying function of the server event service is called which deletes the server event table and an event queue if the client event service is no longer communicating with the server event service.

29. The system as claimed in claim 27, wherein the installation interface is connected to a data capture unit of a technical installation in order to read in events detected by the data capture unit.

30. The system as claimed in claim 27, wherein the server event service has at least one server callback function which can be logged for at least one event and which is called when an event for which it is logged occurs.

31. The system as claimed in claim 27, wherein the server event service has, for every client event service with which it communicates via a communication link, a separate client data record which respectively contains at least one server event table and at least one event queue.

32. The system as claimed in claim 31, wherein the server event service has a tidying function which deletes the client data record if the associated client event service is no longer communicating with the server event service.

33. The system as claimed in claim 31, wherein the server event table is in the form of a hash table and holds data records which contain at least one event identifier and a pointer to a server callback function which is to be logged.

34. The system as claimed in claim 27, wherein the client event service has at least one client logging function for logging client callback functions, at least one client event table for holding data records which describe the log, and at least one request generator for making cyclic requests for event transmission.

35. The system as claimed in claim 34, wherein the client event table is in the form of a hash table and holds data records which contain at least one event identifier and a pointer to a client callback function which is to be logged.

36. The method as claimed in claim 21, wherein events are transmitted by performing the following steps:

detecting, by the installation interface, an event which has occurred and calling the server callback function logged for this event,

producing, by the server callback function, an entry describing the event in at least one event queue,

reading, by the server event service, the entry produced in the event queue upon the next request from the client event service for event transmission,

transmitting the entry via the communication link to the client event service,

receiving, by the client event service, the entry,

ascertaining and calling the client callback function logged for the event, and

executing, by the client callback function, a defined action for the corresponding event in the client application.